

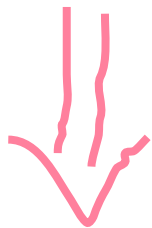
NCS

요구사항확인

first
coding

목차

1. 현행 시스템 분석하기
 - 1-1. 현행 시스템 파악
 - 1-2. 개발 기술 환경 정의
2. 요구사항 확인하기
 - 2-1. 요구사항 정의
 - 2-2. 요구사항의 시스템화 타당성 분석
3. 분석모델 확인하기
 - 3-1. 분석모델 검증
 - 3-2. 분석모델의 시스템화 타당성 분석



<

UML - >

flowchart

요구사항확인

1. 현행 시스템 분석하기

1-1. 현행 시스템 파악

1-2. 개발 기술 환경 정의

- 소프트웨어 요구사항

: HW/SW

OS , , WAS

- 소프트웨어 요구사항

- 시스템이 가져야 할 기능이나 시스템이 만족해야 할 조건을 소프트웨어 요구사항이라고 함

- 요구사항의 분류

- 기능 · 비기능 요구사항 (HW , SW ,)
 - 사용자 요구사항
 - 시스템 요구사항
 - 인터페이스 명세 등으로 요구사항을 분류함

• 요구사항 확인 주요활동

– 소프트웨어 개발단계 (SDLC)

- **요건 정의** → **분석** → **설계** → **구현** → **테스트** → **배포** · **운영** 으로 진행
- 요구사항 확인 절차를 수행하는 단계는 **요건 정의**와 **분석**단계

– 요구사항 확인 주요활동

- **사용자 인터뷰 정리** : ,
- 요구사항 정의
- 요구사항 추적관리
- **기존 업무 분석활동** : .
- **기존 정보 시스템 분석활동**

- **현행 시스템 파악**

(/)

- 현행 시스템 파악 : 사용자 요구사항을 확인 하기 위한 준비단계

- 현행 시스템 파악 주요활동

- 현행 시스템이 어떤 하위 시스템으로 구성되어 있는지 파악

- 제공하는 기능이 무엇인지 파악

- 다른 시스템들과 어떤 정보를 주고받는지 파악 ()

- 어떤 기술요소를 사용하고 있는지 파악 (opensource .)

- 사용하고 있는 소프트웨어 및 하드웨어는 무엇인지 파악

- 네트워크는 어떻게 구성되어 있는지 등을 파악

- 현행 시스템 파악의 목적

- 현행 시스템 파악 목적은 향후 개발하고자 하는 시스템의 개발 범위 및 이행 방향성 설정에 도움을 주는 것을 목적으로 함

- 현행 시스템 파악

- 현행 시스템 파악의 절차

- 1단계

구성, 기능, 인터페이스 현황을 파악하는 단계

시스템 구성 현황 파악

시스템 기능 파악

시스템 인터페이스 현황 파악

- 2단계

아키텍처 및 소프트웨어 구성 현황을 파악하는 단계

아키텍처 파악

소프트웨어 구성 파악

- 3단계

하드웨어 및 네트워크 구성 현황을 파악하는 단계

시스템 하드웨어 현황 파악

네트워크 구성 파악

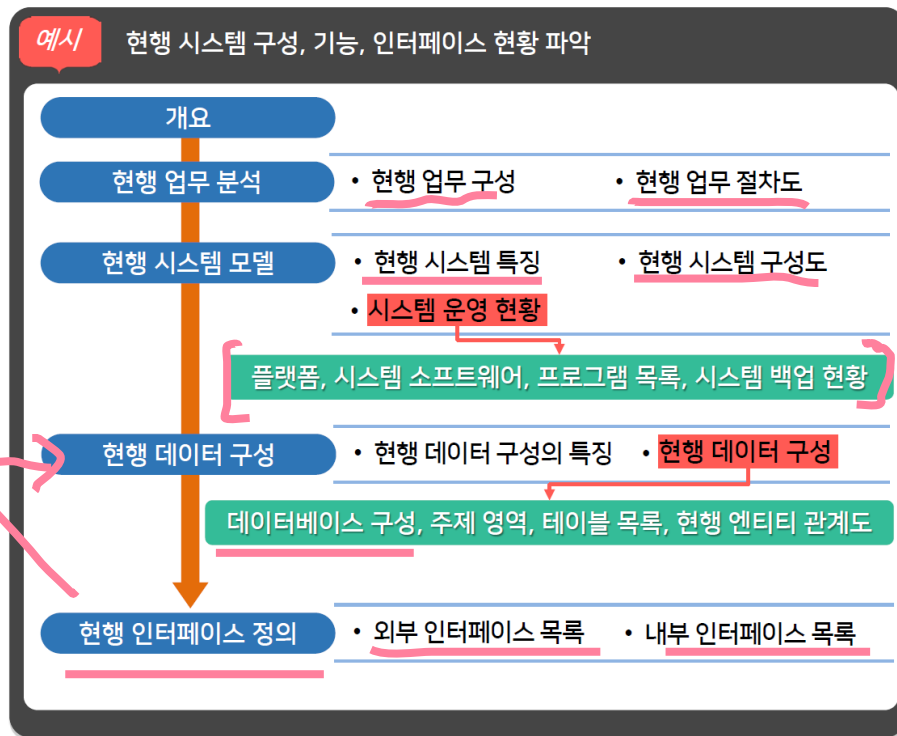
• 현행 시스템 파악

– 현행 시스템 파악의 절차 1단계 : 구성, 기능, 인터페이스 현황을 파악하는 단계

- 시스템 구성 현황 파악
- 시스템 기능 파악
- 시스템 인터페이스 현황 파악

:

Oracle
ms - SQLServer
mySQL



• 현행 시스템 파악

– 현행 시스템 파악의 절차 2단계 : 아키텍처 및 소프트웨어 구성 현황을 파악하는 단계

- 아키텍처 파악
- 소프트웨어 구성 파악

* 아키텍처

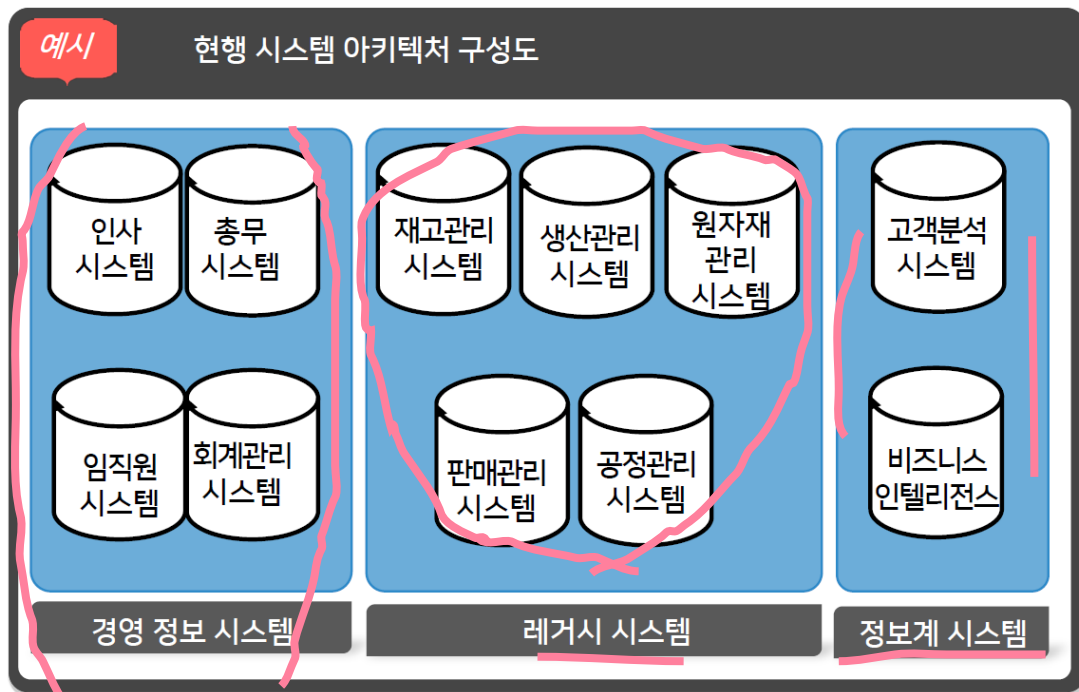
컴퓨터 시스템의

기능(functionality),

조직(organization),

구현(implementation)

에 대한 법칙과 방법을 통칭



• 현행 시스템 파악

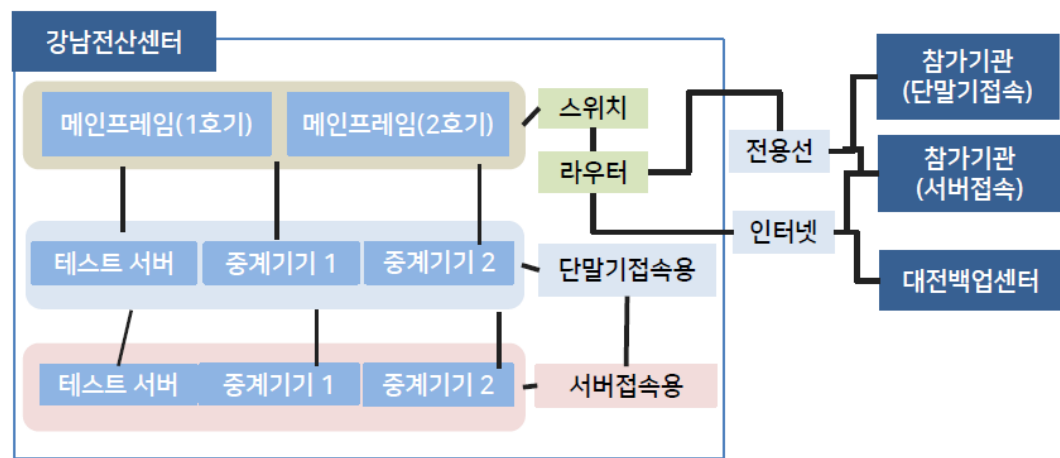
- 현행 시스템 파악의 절차 3단계 : 하드웨어 및 네트워크 구성 현황을 파악하는 단계
 - 시스템 하드웨어 현황 파악
 - 네트워크 구성 파악

* 메인프레임

통계 데이터나 금융 관련 전산업무,
전사적 자원 관리와 같이 복잡한 작업을
처리하는 컴퓨터이다.

예시

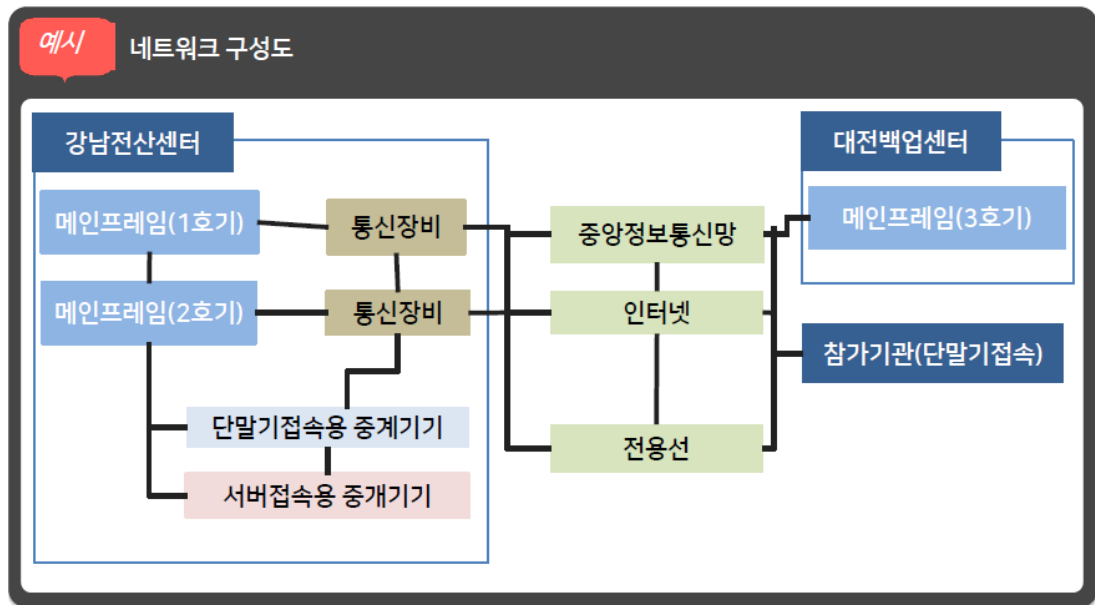
하드웨어 구성도



- 현행 시스템 파악

- 현행 시스템 파악의 절차 3단계 : 하드웨어 및 네트워크 구성 현황을 파악하는 단계

- 시스템 하드웨어 현황 파악
 - 네트워크 구성 파악



- 기술 환경 정의 자료 수집

- 자료 존재 유무 파악

- 기술 환경 정의 자료를 수집을 위하여 수집할 자료의 목록을 정함
 - 자료를 수집하기 위하여 현행 시스템 담당자가 자료와 면담 기록에 필요 자료의 존재 여부를 파악

- 조사 자료 분석 과정

- 조사한 자료를 이용하여 운영체제, DBMS, 웹 애플리케이션 서버 (WAS : Web Application Server) 등을 결정

* 웹 애플리케이션 서버(Web Application Server, 약자 WAS)

웹 애플리케이션과 서버 환경을 만들어 동작시키는 기능을 제공하는 소프트웨어 프레임워크

- 개발 기술 환경 결정

- 시스템 용량 산정

- 분석된 자료를 바탕으로 CPU 용량, 메모리 용량, 디스크 용량을 파악

data



data

- 요구사항 정의서 목표 시스템 구성도 반영

- 용량 산정 결과를 관련자 리뷰를 통하여 수정 및 보정

open source

- 운영체제, DBMS, 웹 애플리케이션 서버 (WAS : Web Application Server), 시스템 용량 산정 결과를 요구사항 정의서, 목표 소프트웨어 구성도, 목표 하드웨어 구성도에 결과를 반영하여 수정

: html

(가)

개발 기술 환경 결정

- 시스템 용량 산정

()

(DB)

Springframework.

WEB WAS를 위한 기초 자료 조사 항목

DATA
DB

항목	설명
시스템 용도 및 서비스 형태	웹 페이지만 제공, 트랜잭션이 빈번하지 않은 웹 서비스 (데이터베이스 연계), 트랜잭션이 빈번한 웹 서비스(데이터베이스 연계)
시스템의 구성 형태	1계층, 2계층, 3계층
접속자 수	평균 접속자 수(24시간 기준), 최고 접속자 수(1시간), 연간 접속자 증가율
사용률	동시 사용자 수, 사용자당 오퍼레이션 수, 이미지 파일과 사운드 파일의 크기, 웹 페이지 크기, 허용 응답 시간
업무 중요도 및 긴급도	중요도(상 · 중 · 하), 긴급도(상 · 중 · 하)
DB 백 엔드 상호 작용 형태	읽기 전용(Read Only), 업데이트(Update), 온라인 트랜잭션 처리(OLTP)
SSL 사용 여부	안전한 통신이 필요한지 여부

http/https

개발 기술 환경 결정

시스템 용량 산정

SW

:

ex) Tomcat ,

WEB / WAS를 위한 기초 자료 조사 항목

항목	설명
운영체제	<ul style="list-style-type: none"> 시스템 구축 예산이 적은 경우(예 2천만 원 이하)에는 유닉스(UNIX)를 도입하기 어려움 리눅스(Linux)는 비용이 저렴하나 유지 관리를 위한 기술 인력을 보유하거나 별도의 계약을 체결해야 함 유닉스(UNIX)는 안정적이고 대량의 처리가 가능하며 기술 지원이 용이하나 비용이 많이 소요됨 윈도우(Windows)는 유지 관리 기술 인력 확보가 용이하고 유닉스(UNIX)에 비해 상대적으로 비용이 저렴하나, 대부분의 대용량 처리 서버에 설치할 수 없음
DBMS	<ul style="list-style-type: none"> 상용 DBMS의 경우 안정적이며 확장성이 뛰어나고 기술 지원을 받기 용이하나 비용이 많이 소요됨 오픈 소스 DBMS의 경우 비용이 저렴하나, 관련 기술을 자체적으로 확보할 필요가 있음 일반적으로 많이 사용되고 있는 DBMS(상용 또는 오픈 소스)를 선택하면 관련 기술 인력 및 기술 자료를 확보하기 용이하고 문제 해결이 용이함
웹 애플리케이션 서버 (WAS : Web Application Server)	<ul style="list-style-type: none"> 표준 규격을 준수하는 웹 애플리케이션 서버(WAS)의 경우 개발용과 운영용을 구분하여 사용할 수 있음 개발용은 가볍고 빠른 오픈 소스 웹 애플리케이션 서버(WAS)를 선택할 수 있음 상용 웹 애플리케이션 서버(WAS)의 경우에는 안정적이며, 대량 처리가 검증되어 있고 기술지원을 받기가 용이함 오픈 소스 웹 애플리케이션 서버(WAS)의 경우 일반적으로 널리 사용하는 웹 애플리케이션 서버(WAS)를 선택하는 것이 바람직함

요구사항확인

2. 요구사항 확인하기

2-1. 요구사항 정의

2-2. 요구사항의 시스템화 타당성 분석

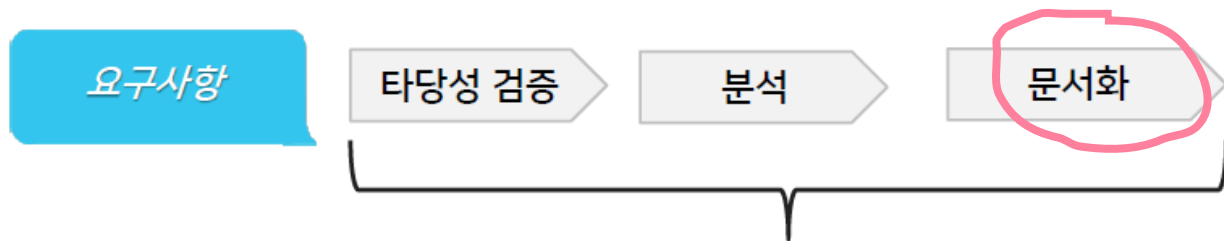
- 요구공학

- 요구공학 정의

- 문서 생성, 검증, 관리하기 위하여 수행되는 구조화된 활동의 집합
 - 요구사항의 획득, 분석, 명세, 검증 및 변경관리 등에 대한 제반 활동과 원칙
 - 요구사항 생성 및 관리를 체계적, 반복적으로 수행하는 활동
 - 요구사항 관리에 포함되는 모든 생명주기(SDLC)활동과 이를 지원하는 모든 프로세스를 포함하는 개념
 - 최종 산출물
 - 요구사항 명세서

- 요구공학

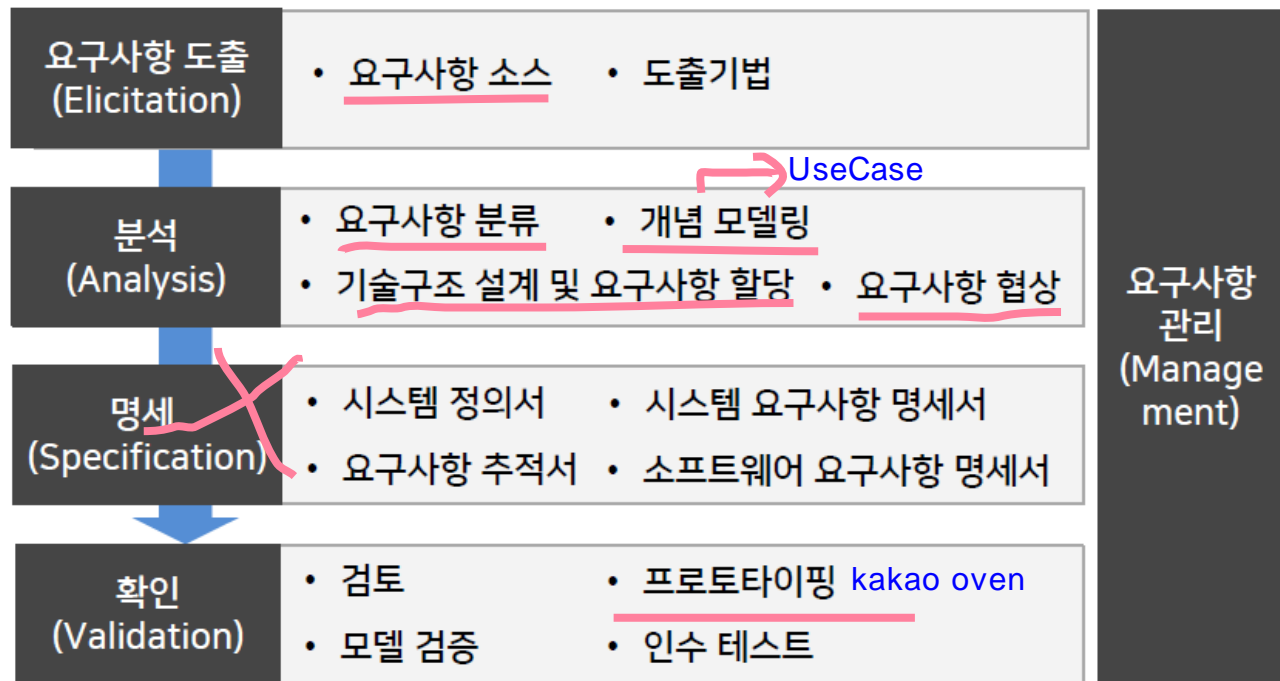
- 요구공학 프로세스(요구사항 개발 프로세스)
 - 요구사항을 명확히 분석하여 검증하는 진행 순서
 - 요구사항 도출, 분석, 명세, 확인 및 요구사항 관리



프로젝트 내내 이를 참고하여 개발하고
최종 목적물과 비교해야 함!

• 요구공학

– 요구공학 프로세스(요구사항 개발 프로세스)



- 요구공학

- 요구공학 프로세스(요구사항 개발 프로세스)

요구사항 도출

- 소프트웨어가 **해결해야 할 문제**를 이해하는 **첫 번째 단계**
- **요구사항이 어디에 있고, 어떻게 수집할 것인가를 찾는 행위**

이해관계자(Stakeholder) 식별

개발팀과 고객 사이의 관계 형성

*다양한 이해관계자와의
효율적인 의사소통이 중요!*

요구사항 분석

- 요구사항들 간 상충되는 것을 해결
- 소프트웨어의 범위 파악
- 소프트웨어가 환경과 어떻게 **상호작용**하는지 이해

 시스템 요구사항을 정제하여 소프트웨어 요구사항을 도출!

- 요구공학

- 요구공학 프로세스(요구사항 개발 프로세스)

요구사항 명세

- 체계적으로 검토, 평가, 승인될 수 있는 **문서를 작성하는 작업**
- 시스템 정의, 시스템 요구사항, 소프트웨어 요구사항 작성

요구사항 확인

- 분석가가 요구사항을 이해했는지 **확인(Validation)** 필요
- 요구사항 문서를 **검증(Verification)** 하는 것이 중요

- ✓ 회사의 표준에 적합한가?
- ✓ 이해 가능한가?
- ✓ 일관성이 있는가?
- ✓ 완전한가?

- 이해관계자들이 문서를 검토해야 하고, 요구사항 정의 문서들에 대해 **형상관리**를 해야 함 **git** 가 .
→ 일반적으로 **요구사항 관리 툴** 이용
- 리소스가 요구사항에 할당되기 전에 문제를 파악하기 위하여 **검증을 수행**

- 요구사항 분석 기법

- 요구사항 분석 기법의 종류

✓ 요구사항 분류(Requirement Classification)

✓ 개념 모델링(Conceptual Modeling) UseCase

요구사항 할당(Requirement Allocation)

요구사항 협상(Requirement Negotiation)

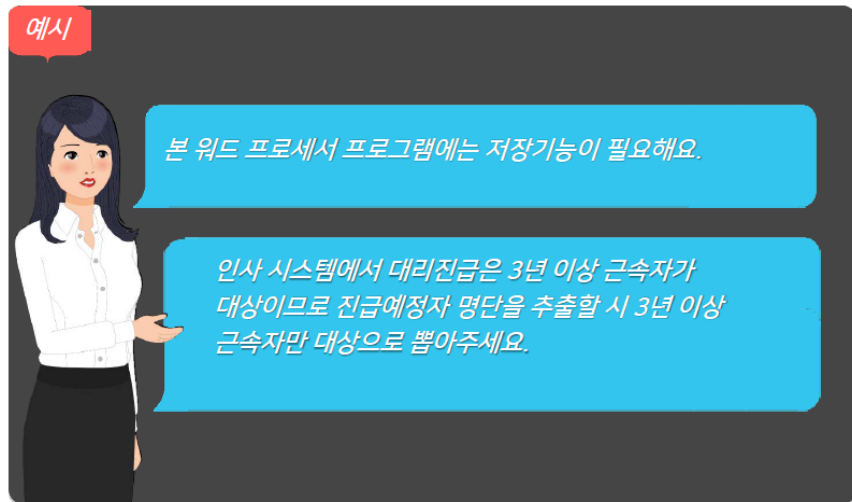
정형 분석(Formal Analysis)

• 요구사항 분석 기법

– 요구사항 분석 기법의 종류 : 요구사항 분류(Requirement Classification)

• 기능적 요구사항

- 시스템이 어떤 기능을 갖추어야 하는지를 요구하는 사항
- 시스템 기능을 정의한 것 → 시스템이 동작하는 내용에 대해 정의한 것



• 요구사항 분석 기법

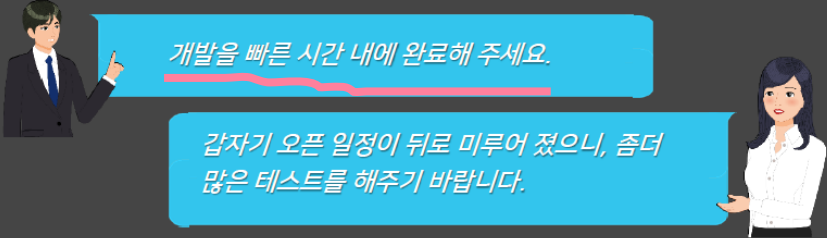
– 요구사항 분석 기법의 종류 : 요구사항 분류(Requirement Classification)

• 비기능적 요구사항

– 기능적인 부분 이외의 요구사항


– 시스템 전체적인 특성 → 시스템 성능, 신뢰성, 확장성, 운영성, 보안 등과 관련된 요건

예시 신뢰성, 사용성, 효율성, 보수성, 이식성, 장애 억제성, 효과성, 운용성, 기술 요건, 가용성, 성능 확장성, 운용·보수성, 보안, 환경 조건



개발을 빠른 시간 내에 완료해 주세요.

갑자기 오픈 일정이 뒤로 미루어 졌으니, 좀더 많은 테스트를 해주기 바랍니다.

 실제 프로젝트 완료 후 시스템 사용자의 만족도 측면에서 **비기능적 요구조건**도 매우 중요한 반영요소가 됨

• 요구사항 분석 기법

– 요구사항 분석 기법의 종류 : 개념 모델링(Conceptual Modeling)

- 문제에 대한 모델링 → 소프트웨어 요구사항 분석의 핵심
- 모델은 문제가 발생하는 상황에 대한 이해를 증진시키고 해결책을 설명
→ 개념 모델은 문제 도메인의 엔터티(Entity)들과 그들의 관계 및 종속성을 반영!

=

* 모델(model)

특정 리소스를 이용해서 특정 결과를 도출하는 방법, 규칙,과정을 정의해 놓은 것

* 도메인

일반적인 요구사항이나 전문 용어 또는 컴퓨터 프로그래밍 분야 등에서 문제를 해결하기 위해 설계된
어떤 프로그램에 대한 기능성을 정의하는 영역
(예) 광고회사의 광고와 관련된 지식 = 도메인

• 요구사항 분석 기법

– 요구사항 분석 기법의 종류 : 개념 모델링(Conceptual Modeling)

- 문제에 대한 모델링이 소프트웨어 **요구사항 분석의 핵심**

- 개념 모델의 종류와 표기법

- 유스케이스 다이어그램(Use Case Diagram) :

?

- 데이터 흐름 모델(Data Flow Model)

- 상태 모델(State Model)

- 목표기반 모델(Goal-Based Model)

- 사용자 인터랙션(User Interactions)

- 객체 모델(Object Model) - >

:
(,)(,)

- 데이터 모델(Data Model)

- > DB(RDB)

- 대부분의 모델링 표기법은 UML(Unified Modeling Language)을 사용!

• 요구사항 분석 기법

– 요구사항 분석 기법의 종류 : 요구사항 할당(Requirement Allocation)

- 요구사항을 만족시키기 위한 아키텍처 구성 요소를 식별하는 것
- 다른 구성 요소와 어떻게 상호 작용하는지 분석을 통하여 추가적인 요구사항을 발견 할 수 있음.

* 아키텍처

컴퓨터 시스템의

기능(functionality),

조직(organization),

구현(implementation)

에 대한 법칙과 방법을 통칭

• 요구사항 분석 기법

– 요구사항 분석 기법의 종류 : 요구사항 협상(Requirement Negotiation)

- 두 명의 이해관계자가 서로 상충되는 내용을 요구하거나, 요구사항과 리소스, 기능과 비 기능 요구사항들이 서로 상충되는 경우 어느 한 쪽을 지지하기보다는 적절한 트레이드오프 지점(절충점)에서 합의하는 것이 중요
- 요구사항에 우선순위를 부여하여 중요한 요구사항을 필터링하고 요구사항들 간 상충되는 문제를 해결할 수 있음

- 요구사항 분석 기법

- 요구사항 분석 기법의 종류 : 정형 분석(Formal Analysis)

- 요구사항을 기술하여 분석하는 방법을 수학과 논리학에 기반을 두어 자연언어가 내포하는 애매모호함이나 불확실성을 제거하는 분석 기법
 - 정성적 요소보다는 정량적, 구체적으로 기술하여 명세화 함 !
 - 추후 정확하게 측정 가능한 요소로 요구사항을 명세화 하지 않으면 시스템 요청자와 구축 수행자 간의 최종 인도 시 분쟁의 소지가 많음

요구사항확인

3. 분석모델 확인하기

3-1. 분석모델 검증

3-2. 분석모델의 시스템화 타당성 분석

- 요구사항 분석활동

- 요구사항 분석활동

- 도출된 각각의 요구사항을 좀 더 풍부하고 상세하게 이해하고 이를 여러 가지 방법으로 표현
→ 요구사항을 분류하고 조직화하여 명세를 구체화해 나가는 것
 - 기능분할
 - 시스템 관점에서 외부에 제공해야 하는 서비스(기능)를 유형별로 최하위 단위 구성요소에 도달할 때까지 분할
 - 시나리오 기반 분석
 - 시스템 외부의 사용자 / 이해관계자 / 주변 환경이 시스템과 긴밀하게 상호작용하며 시스템을 사용하는 시나리오 작성
 - 유스케이스(Use Case) 분석도 시나리오 분석에 해당
 - 유스케이스 분석

()

- 요구사항 분석활동

- 유스케이스 분석

- 대표적인 시나리오 기반 요구사항 분석 방법으로 활용
 - 사용자 요구사항 문서와 같은 입력 자료로부터 이해관계자와 관심 대상인 시스템 간의 상호작용 식별
→ 유스케이스 다이어그램으로 작성

* 유스케이스 다이어그램 (Use Case Diagram)

시스템에 대한 이해관계자인 액터와 상호작용을 나타내는 유스케이스 간의 관계와 구조를 가장 개괄적으로 보여주는 도표

요구사항 분석활동

유스케이스

- 시스템 경계 밖에 위치한 액터가 특정 목적을 달성하기 위해 시스템이 제공하는 기능을 이용하여 시스템과 주고받는 일련의 상호 작용을 시나리오로 나타낸 것

유스케이스 모델링

기능 요구사항 후보 목록과 자료로부터 다음의 순서로 모델링



액터(Actor)

- 유스케이스를 사용하는 사람만을 의미하는 것이 아니라 유스케이스 관련이용자와 시스템을 의미

유스케이스 (Use Case)

- 모델화 대상이 외부에서 제공하는 서비스
→ 액터가 이용하는 서비스

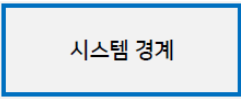

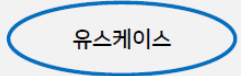
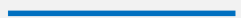
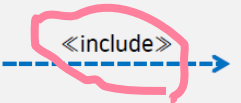
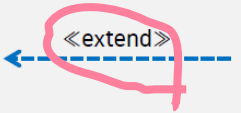
관계 정의

- 관련된 액터와 유스케이스를 연결한 선
- 관계 : 액터와 유스케이스의 관련성
(유스케이스 행위자 표시가 아님)

시스템 경계

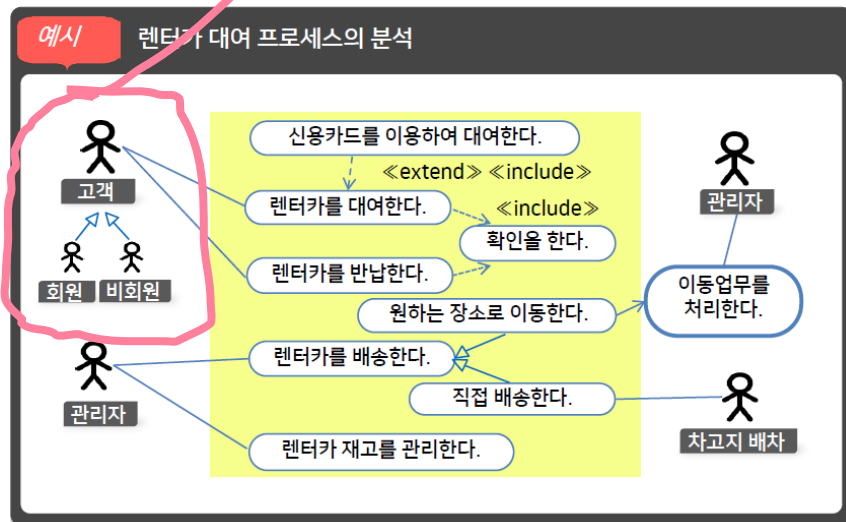
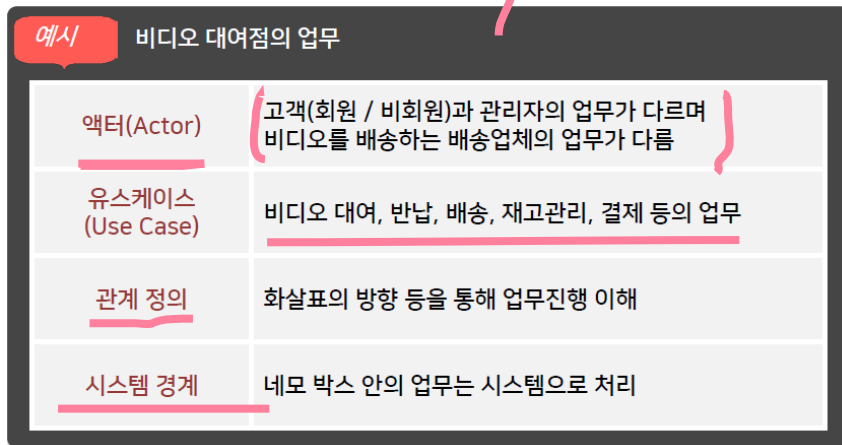
- 시스템화 대상 지정
- 시스템화할 대상과 시스템 대상범위가 아닌 것을 구분하기 위하여 표시

- 요구사항 분석활동
 - 유스케이스 표기법

요소명	표기	설명
시스템 경계		시스템의 범위 표기
액터		시스템과 상호작용하는 시스템 외부 객체
유스케이스		시스템에 의해 수행되는 행위 또는 기능
연관		액터와 유스케이스 간의 상호작용 관계
포함		하나의 유스케이스가 또 다른 유스케이스를 사용하는 것을 나타내는 두 유스케이스 간의 관계
확장		하나의 유스케이스가 또 다른 유스케이스에 행위 또는 기능을 추가하는 것을 나타내는 두 유스케이스 간의 관계 ()
일반화		액터와 액터, 유스케이스와 유스케이스 간의 관계로 두 개체 간에 일반화 관계가 있음을 나타냄

가 가 .

- 요구사항 분석활동
 - 유스케이스 표기법



• 분석모델 검증방법

– 분석모델 검증 절차

- 정보시스템의 분석설계가 잘 도출되었는지 검증하는 절차

단계	검토대상
① <u>유스케이스 모델 검증</u>	<u>73</u> <u>능력</u> <u>액터 / 유스케이스 / 유스케이스 명세서</u>
② <u>개념 수준의 분석클래스 검증</u>	<u>클래스 도출</u> / 클래스 명과 속성 / 클래스들간 관계
③ <u>분석클래스 검증</u>	스테레오 타입 / 경계 및 제어 클래스 도출 / 관계 및 상세화 정도

- 분석모델 검증 완료 후 검토의견을 작성하고 공유함

• 분석모델 검증방법

– 유스케이스 모델 검증

점검대상	점검내용
액터	<ul style="list-style-type: none"> • 기능 구현에 관계되는 액터가 모두 도출되었는가? • 액터 목록에서 액터명이 역할 중심으로 명명되었는가? • 요구사항 정의서, 요구사항 기술서에 외부 / 내부 액터가 모두 도출되었는가? • 액터 목록과 액터 명세서에 기록된 액터가 타당한가?
유스케이스	<ul style="list-style-type: none"> • 요구기능 구현에 필요한 유스케이스가 모두 도출되었는가? • 도출된 유스케이스를 논리적으로 연결하여 누락된 기능 파악 • 도출된 유스케이스가 유스케이스 목록과 유스케이스 명세서에 반영되었는지 확인 • 도출된 유스케이스의 논리적인 합이 과업 범위와 일치하는지 비교 • 도출된 유스케이스들이 논리적으로 그룹화되었는지 확인 <ul style="list-style-type: none"> - 그룹화는 액터 기준, 연관 관계 기준, 동시성 기준이 가능 • 유스케이스 기능 범위가 다른 유스케이스 기능 범위와 중복되는지 확인
유스케이스 명세서	<ul style="list-style-type: none"> • 유스케이스 명세서 형식에 중요 항목이 누락되지 않았는지 확인 <ul style="list-style-type: none"> - 사전 및 사후 조건, 주요 흐름, 서브 흐름, 예외 흐름 등 • 유스케이스의 주요 이벤트 흐름이 모두 도출되고 논리적으로 타당한지 확인 • 유스케이스를 구현하기 위하여 필요한 <u>입출력 항목이 모두 도출되었는지 확인</u>

- 분석모델 기술적 타당성 검토

- 분석모델 기술적 타당성 검토 절차

- 요구사항에 대한 기술적 타당성 검토 절차와 동일한 방식으로 분석모델이 기술적으로 타당한지 검토함

- 타당성 검토 내용

- 성능 및 용량분야
 - 시스템간 상호 운용성 분야
 - 시장 성숙도 및 IT 트렌드 부합성 분야
 - 기술적 위험분석 분야

• 분석모델 기술적 타당성 검토

– 타당성 검토 내용

• 성능 및 용량분야

– 성능평가 확인

S/W = > :

- » 분석모델이 최종 사용자 응답시간에 대해 요구사항이 도출되었는지 확인하고 적정성을 검토
- » 시스템 개발 이후 최종 사용자 응답시간에 대한 사용자 불만족 사항을 사전에 차단하는 것이 1차적인 목적임
- » 요구사항 도출 이후 분석한 모델이 개발 시스템의 특성과 구축비용에 따라 응답시간이 달라질 수 있으므로 적정성을 검토하는 것이 목적임

• 분석모델 기술적 타당성 검토

– 타당성 검토 내용

• 성능 및 용량분야

– 용량산정 확인 → 가용성과 확장성 측면으로 검토

- » 분석모델이 시스템의 업무 특성에 맞도록 적절하게 도출되었는지 확인하고 이에 대한 적정성을 검토
- » 시스템 운영 시 가용성에 대한 사용자의 기대치를 분석하였는지 검토
- » 가용성 목표가 업무특성과 경제성을 고려하여 적정하게 도출되었는지 검토
- » 시스템 확장성에 대한 분석모델이 도출되었는지 확인하고 적정성 및 경제성을 검토
- » 하드웨어나 네트워크 장비의 경우 내부 자원의 추가로 인한 수직적 확장과 동일 장비의 추가로 인한 수평적 확장을 고려할 수 있음 → 각 기반 요소 별로 효율적인 확장방안이 적용되었는지 검토
- » 목표 시스템에 대한 아키텍처가 분석모델에 맞게 설계되기 위해 필요한 절차를 수행했는지 검토
- » 시스템 확장성에 대한 분석모델이 업무량의 증가 및 특성, 업무범위의 확대 등을 고려하여 적정하게 도출되었는지 검토

- 분석모델 기술적 타당성 검토

- 타당성 검토 내용

- 시스템간 상호 운용성 분야

- 시스템의 유연성 측면에서 분석모델이 충분히 도출 되었는지 확인
 - 서로 다른 시스템간 상호 정보 및 서비스 교환 가능성 등을 검토
 - 제안 요청서, 제안서 등에 명시된 시스템 유연성에 관한 내용이 분석모델에 적절하게 도출되었는지 확인하는 것이 목적
 - 시스템의 확장, 변경, 교체 등 시스템 구성 환경의 변화에 대하여 응용 시스템이 유연하게 적응할 수 있는지 검토
 - 하드웨어 및 패키지 공급자에 대한 종속성을 최소화하고 환경 변화에 대응할 수 있는지 검토

- 분석모델 기술적 타당성 검토

가
가

- 타당성 검토 내용

- 시장 성숙도 및 IT 트렌드 부합성 분야

- 시스템 구축 시 요구되는 영역별 정보 기술들의 시장 성숙도 및 발전 방향을 파악

- 도출된 요구사항이 IT 시장 성숙도 및 트렌드에 부합하는지 판단해야 함

- 시장 성숙도가 낮거나, 발전 방향에 부합되지 않는 기술들은 향후 더 이상 사용되지 않을 가능성이 높아 시스템의 유지보수가 어려운 상황이 발생할 가능성이 높음

- 분석모델 기술적 타당성 검토

- 타당성 검토 내용

- 기술적 위험분석 분야

- 분석모델을 만족시키기 위하여 적용한 기술의 복잡성, 검증 여부, 의존성 등에 대하여 위험 발생 가능성, 영향도를 파악

- 분석모델 타당성 의견 작성

- 분석모델 타당성 의견을 작성하기

- 분석모델까지 요구사항 추적표를 작성
 - 타당성 검토 의견 컬럼을 추가
 - 작성된 요구사항 추적표에 타당성 검토 의견을 작성

- 분석모델 타당성 의견 공유

- 타당성 분석 결과를 관련 이해관계자가 검증한 후 관련 이해관계자 검증을 거친 타당성 분석 결과를 확인하고 배포 및 공유